DOCUMENT RESUME

ED 241 940 CS 208 201

AUTHOR Evans, Ron
TITLE Taking a Hard Look at Software. What about Wimpy Software?
PUB DATE Nov 83
NOTE 17p.; Paper presented at the Annual Meeting of the National Council of Teachers of English (73rd, Denver, CO, November 18-23, 1983).
PUB TYPE Speeches/Conference Papers (150) -- Viewpoints (120) -- Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Artificial Intelligence; Authoring Aids (Programing); *Computer Assisted Instruction; *Computer Software; *English Instruction; Language Arts; Literature Appreciation; *Microcomputers; Program Improvement; Reading Comprehension; *Teacher Developed Materials; *Teacher Role; Word Processing; Writing Instruction; Writing Skills

ABSTRACT

Much of the computer software currently available for English teachers fails to assess adequately computer strengths and weaknesses. Labeled "wimpy software," these products are often little more than animated textbooks whose lesson formats exercise little higher-order reasoning. The future for good quality software, therefore, rests with English teachers themselves. By using existing authoring systems and word processor programs, English teachers can create their own lessons. Apple's Super Pilot, for example, presents a respectable menu of options for presenting interactive lessons, for tallying results, and for augmenting these lessons with graphics in a range of colcrs. With this disk package, teachers can create lessons in grammar and usage, reading comprehension, simple composition skills, or even literary backgrounding. Word processing programs such as the Bank Street Writer make it easy to compose lessons in free writing story completions, creating suitable writing styles, word substitutions, definitions, and sentence expansions and combinings. Another potential area of computer use may be seen in composition tutorial programs that use artificial intelligence techniques to develop worthwhile interactivity between writer and computer. With these three aids English teachers can learn to create their own floppy-disk factory. (HOD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED241940

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
X This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

Taking a Hard Look at Software
Ron Evans
Professor of English
University of West Florida

What About Wimpy Software?

The industry grows so fast that it tumbles over itself, rushing to get new product diskettes on the market, not unlike the department store customers at a one day clearance sale groping for position at a cashier's desk. Why not? The payday for a computer wise person can bourgeon forty hours of programming into thousands of profit dollars for any enterprising instructional designer. So it goes in the floppy disk business of instructional software.

But why then, have many English teachers in particular found disappointment in so many of the engagingly advertised software packages? The answer is complex, at lesat threefold. First, many English teachers don't realize the strengths and weaknesses of the microcomputer in a language-instruction mode. Second, some English teachers have already mastered some programming proficiency, thereby being technically able--perhaps even "spring-loaded"--to critique the failures of simplistic software. Third, the needs of secondary English programs have not really been addressed in any systematic way for a computer-assisted instructional designer.

Certainly, there is a plethora of linear programs, that present text booklike exercises in spelling, vocabulary, parts-of-speech, and even simple-sentence recognition. Some are even augmented with clever graphics to divert the monotony of print displays. But the problem remains that the content of these programs suits basic instruction more than advanced. In terms of J.P. Guilford's departments of intellect (1) most software relies on memory and low-level convergent responses from users. For example, on one program from the Minnesota

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Ronald V. Evans
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

CS 208 201

Education Consortium, the teacher is presented with a menu of 11 rather interesting word games with names like "Amazing," "Spell," and "word find." Each game requires the teacher to enter words familiar or pertinent to student's work. But each game merely presents words, memory cues, and immediate looping feedback to the same word for incorrect responses.

Another typical floppy-disk program from Micro-Power and Light Co., called "Sentences" leads the learner through a series of directions about how to move through the lesson using the space bar, augmented by both some interesting visual graphics for tilting and boxing emphasize elements and some cute digital sounds of calliope music and tapping as letters or asterisks appear on the display screen. The series of lessons on the diskette covers simple subjects, predicates and sentence fragments. A "user-friendly" device in the program allows the display screen to address the learner by name at the outset, and during each reinforcing response, or branch, made by the program after learner responses. For variety, this program offers a "baseball game" in which the learner selects a single, double or home run degree of difficulty and then must type in a highlighted word-choice for simple subject. This highlighting makes use of a programming feature called "text-porting" which we will discuss later in this paper. At any rate, the program proceeds along a predictable route, first cueing the learner by defining and illustrating, then offering multiple-choice responses, followed by reinforcements or repetitions respectively for correct or incorrect responses.

It is an interesting program, in the main, popping logically-sequential lessons with flashing elements to engage the learners.

But alas! It's a dull program for anyone beyond 8th grade. I suspect even for remedial students, except for its tangent value as a reading-instruction device, emphasizing main ideas and predicative details. Furthermore, the branches are very primitive, merely allowing the learner to remain at the multiple-choice selection menu until making the correct choice. These are examples, then, of "Wimpy Software." And even though they may not be representative samples of all current English-topical programs, they do illustrate the fact that such products are little more than animated textbooks, whose pages are turned for you but whose lesson formats exercise little higher-order reasoning.

My feeling is that such programs as these fail to assess adequately the strengths and weaknesses of the computer. After all, the presentation of learning material and the evaluation of learner responses to that material is so fundamental to all instruction, that one almost demands the computer do better than what's currently available. Naturally, the true strengths of the microcomputer will evolve with the programming skills of experienced English teachers, who wish to legitimately include this wonderful machine in the process of the class. Optimistically, such an evolution will be enhanced by the use of authoring systems like Apple's "Super Pilot," together with planning of programs through computer-simulation pages in ways I described last year on a similar panel in Washington D.C. Meanwhile, there's plenty to do right now.

Remember also that the current "Wimpy Software" problem has been touched off, at least in part, by what I call the "techno-clic" hysteria of microcomputers in classrooms. We, the staid ol' English

teachers, known for our stalwart defense of Standard English usage, should move slower than the others in the herd. What we produce for our classes should be just that, for our classes. If these programmed lessons get good results, they should be explained and [illegible]stratively shared through meetings like these. Enough could be held back, through, so that the entire program can be marketed if it's good.

The "techno-chic" hysteria now accounts for microcomputers' being in 13,000 of the nation's 16,000 school districts, according to AP education writer Lee Mitgang. More than 85% of all high schools are averaging 10 or 11 compute s each. But as is commonly known, the machines are being only minimally used, especially by language-arts and English teachers. Notwithstanding the dramatic claims by Seymour Papert, author of Mindstorms, the extended contributions of micro-computer-assisted instruction seem vague in contrsat to their similar success with basic instruction: Apparently, smaller segments of language, with fixed rightness and wrongness, ARE BETTER SUITED TO CURRENT COMPUTER-DELIVERED LESSONS. Hence, we get word-centered activities with low-level convergent and memory questions. (2)

This spawns the critical insight that higher-order language learning is not happening with current software. Pappert's LOGO program, for instance, helps intermediate elementary students with logical thought, but doesn't provide them with the language structures to express that thought. A recent edition of College Board News warns that the commercial floppy disk progarms for SAT score improvement may not be generically beneficial, despite claims and money-back guarantees. Even New York"s Bank Street College of Education has found little intermediate elementary schoolers are improved, by computers, in the way they learn or perform other tasks. In short, there is every reason

in the electronic world for English teachers to take their time and to produce carefully wrought microcomputer programs that will genuinely enhance reading, writing and verbal reasoning.

Well then, what kinds of programs might a person produce, that are both easy enough for teachers to do and challenging enough for secondary English students to work on? First of all, everyone interested in computers and English ought to seek compatible exercise formats from textbooks. These exercises should be transferrable to computer-delivery with minimum changing. They should demand more of the learner than just memory-recall or simple convergence. They should be designed with "user friendly" attributes--personalizing gimmicks like addressing by name, asking opinions or feelings about these lessons. They should include genuine interactivity between program and learner, meaning that branches for both correct and incorrect responses should be worthwhile and constructive. For instance, just saying "super" or "good work" as a reinforcement is not as good as adding a summary of what the previously selected correct response was and why it was correct.

"Good work, Igor, you figured out that the subject of a sentence can be more than one word."

Incorrect responses should automatically branch to similarly-worded facsimiles of the original stimulus, not merely a replay. These branches should also match a degree of difficulty with the degree of wrongness found in the incorrect responses. Multiple-choices, for instance, can be so written that one is only slightly wrong, a second more obviously wrong, and a third grossly wrong. A learner choosing either the second or third distractor, therefore, should be required to do two or three times the remedial practice, that the learner does

who chooses the first incorrect distractor choice. Finally, visionary homemade software should make use of graphics, game formats, or computer gadgetry, which will engage the learner and indirectly demonstrate elements of computer literacy and program capability.

For the last two suggestions, it would be well for teachers to get familiar with existing "Authoring Systems" and word-processor programs. As has been suggested by my presentation in D.C. last year, and by Wallace Judd in the October Kappan.
authoring systems are enormously helpful in guiding computer-neophyte English teachers. By using these floppy-disk devices, the teacher is relieved of having to know a great deal about programming. Everything necessary for a good, sound computerized lesson is on the authoring disk. Though many serious programmers have criticized authoring systems for this very quality, I contend that their criticisms betray their biases as an advanced-skills elite, not addressing the needs of the ordinary secondary English teacher who is trying to make use of the computer in original and novel ways.

Apple's Super Pilot presents a respectable menu of options for presenting interactive lessons, for tallying results and for augmenting these lessons with graphics in a range of colors. Also it is possible to highlight key terms inside colored boxes called text porting.

Though somewhat intimidating at first, this relatively inexpensive disk package can be mastered over a weekend of concentrated computer play. Lessons in grammar and usage, reading comprehension, simple composition skills, even literary backgrounding or interpretation

can be produced and tried out on volunteer students. Naturally, these early weekend products will undisguisedly reflect the textbooks then being used in the class. But that's no bad thing! When a teacher is starting out on the computer, using an authoring system, there is enough complexity right there! Some pirating of well organized material from a familiar source is understandable. The difference is that this will be a developmental stage for the English teacher and not a final commercial product, as some of the wimpy software described earlier.

Similar to the authoring disk in its usefulness is the word-processing program. I, for instance, am completely capitivated by Bank Street Writer from Scholastic, because its streamlined simplicity makes it easy to compose lessons on, for the teacher, and easy to teach students to use when doing the lessons. I won't bore you with specific details about how it works. That you can find out for yourselves on a 30 day free trial. But I do wish to explore and to amplify the kinds of lessons that are possible with this word-processor.

In the User's Handbook, accompanying Bank Street Writer, are twenty-one ideas for using the word processor both to present lessons and to guide students in using word-processor functions in doing these lessons. In effect, the activity is designed to kill two birds with one stone--to familiarize kids with this neat tool and to generate either creative writing or editorial skills. These lessons include free writing until the screen is full, story-completions, suitable style (rewriting for), word substitutions, definitions, sentence expansions and combinings. To these novel instructional adaptations of the word processor can be added a variety of other exercises, suited to either the WRITE OR THE EDIT mode respectively.

In the WRITE mode, for example, the general idea is to use minimal cues to set up the kinds of writing desired. For example, with both narrative and expository models, lead-in sentences could be pre-written on the displays, followed by transition words or phrases heading up blank spaces. Based on the main idea found in the cue sentence, coupled with the sequential hints of the transitions, the learner types in "follow-up" sentences which show up on the display previewing to determine how much sense is being made between the cue sentences, the transitions, and the learner's sentences. For example, a simple narrative exercise would provoke the learner with a dramatic opener like "night fog wrapped around Marva like puffs of cold steam, as she waited for Jason near the family crypt in a deserted part of the cemetary." And then . And then . And then . Finally, . Neither Jason nor Marva had considered this turn of events. So ." If they choose, students are encouraged to add more anti-climatic events than are cued by the "And then's," as well as to expand the "Finally" cued climax statement. The summary sentence may also be explained as either an end to a brief sketch or as a transition to even more plot events if desired. Further, the summary sentence can be so arranged that it will only appear after the learner has typed in his anti-climax sentences, thereby creating a kind of natural priority list for story writing.

Expository exercises follow the same pattern as the narrative, only expository transitions would be used, such as "on the other hand," for contrast; "Thereafter," for sequence; "For instance," for illustration; etc. A major cue sentence like "Television has changed the lifestyle of Americans" could be followed by "For example" .

"Furthermore," "By this I mean," etc. After the sentences are typed in, a summary sentence would then appear on the diaplay, "All in all, American viewing habits have created a variety of other habits, "which provides a kind of check for the learner to see how his or her sentences relayed the meaning of the first and last models.

In all such exercises, it would be best to write-protect the floppy disk when exercises are written, so that they may not be erased. Also you should require students to run a "hard copy" of their final work after they have thoroughly checked it on the display screen. This way, the pre-writing and final drafting are assured and the turn-in copy should be worthwhile for intensive grading or comment. Naturally, it will be necessary for students to learn basic EDIT mode functions so that they can refine the final display before running the printer. The teacher may program further instructional exercises, however, using the EDIT mode exclusively, such as whole paragraphs each needing punctuation, capitalization, spelling, verb-pronoun agreement, tense, subordination, sentence-combining or parallelism. In fact, an entire sequence of EDIT mode lessons could be programmed, each concentrating, in turn, on one of the topics mentioned, followed by 3 or 4 mixed-bag paragraphs in which appear samples of all mistakes. Again, a printout copy will be turned in for the teacher to grade and retain in a folder.

Beyond the authoring system, the Word Processor and the cue-closure technique, there is still one more vital area where computers can contribute valuable software to English teaching. Composition tutorial programs, as they may generically be called, are pioneer attempts to use artificial intelligence techniques to develop worthwhile

interactivity between writer and computer. Perhaps such packages are best exemplified by Bell Labs mainframe program called "Writer's Workbench," a tutorial program currently in use at Colorado State University. This really marvelous CAI service receives through keyboard samples of student writing, the proceeds to analyze it for style, organization and usage features. Clearly, such capability demands greater storage and processing than most microcomputer systems provide. But it is still possible to make smaller scale tutorials that can be both helpful for young writers and challenging for young computer-literate students as they explore the comparative "thought processes" of man and machine.

To make the software for a writer's tutorial, one must first plan the functions in pseudocode and then proceed to program these functions in either BASIC or PASCAL, depending on micro capability. The priority functions should be a broad lexicon (or dictionary) and a sentence parser. With these two data files in memroy, the microcomputer can process most sentences typed in through the keyboard. With increased skill and understanding in computers, some teachers could even add to the program key style comments concerning easily quantified factors like monotonous sentence patterns, readability involved with sentence length and internal complexity of structure, variation of sentence openers and modifier placement, and ratios of adjective to noun, nouns to verabs, or subordinate to main clauses.

Once a writer has typed in a 200 word sample of prose, the computer print out would first display a complete sentence passing, which the student could check for accuracy, since virtually no computer

can accurately parse every single word in sentence contexts. From this activity, the student would glean the importance of human monitors and the flaws of the sentence calculus, using structural grammar principles in a series of "If-Then" commands. For instance, the program would read, "If word equals axion the, then word-plus-one equals noun." Or this could be expedited by having programmed the part of speech into the lexicon data file. Thus, the direction would read, "If article, then word-plus-one is noun," or "If pronoun and word plus one-preposition, and word plus two-article, then word plus three-noun," etc.

Currently, there are available several major lexicons, some of which give semantic categories kind of conceptual classes. These may be adapted into microcomputer data files, to permit the computer to feedback interpretive comments like "your essay seems to be about social control . . . or political systems . . . or values clarification."

The worthwhileness of writer-tutorial programs on microcomputer has yet to be proven. During prewriting, I feel that young writers, learning control over usage and coherence, will benefit greatly from the immediate feedback offered by such a program. Studying the printouts of their typed-in samples provides students with such a variety of analytical commentary that they will be hard pressed not to glean something out of it. In one long look, each student can be informed about whether his sentences are monotonously structured; if too long or short; if the number of main exceeds subordinate ideas; and perhaps most important, whetehr the main concept of the passage is sustained throughout. Moreover, such a printout can focus the teacher's interpretive comments during post-writing discussions.

At the University of West Florida, we are currently working on such a program. Painfully, one-step-at-a-time, we are debugging the parser, expanding the lexicon to include up to 10,000 words, and working out the style comments based on the quantifications made by the parser and lexicon. Further, we plan to develop true interactivity betewen the program and the learner, so that the preliminary style commentary will be followed by a menu of options for the learner to choose from. Each option will concern one aspect of the laws the computer finds in the prose. For instance, a learner choosing sentence structure from the menu, would receive a display of his own sentences in which subordination parallelism or modifier placement could be improved.

The future for software, in short, lies in the hands of English teachers themselves. Only they know what they need, and only they can test their programming in their own classes so easily. Sure, there are lots of advances being made commercially, like Scholastic's new subscription programming service--Microzine "magazine." These will offer things classroom teachers can't do. But the journeyman stuff for computereducation reposes with the teachers. With aids like authoring system and word processing program, each school can become a veritable floppy-disk factory. Agents from big money publishing or merchandising houses will begin to sense this and establish regular "routes" for buying computer softwares. Better yet, regional consortiums, like the one in Minnesota, can be expanded throughout the country so that all profits from the sales of good software can roll back for local benefit. The future is now, and it's only a disk away.

(Organization's Name)
(Organization's address)

Dear :

(body of letter)

Sincerely,
(Your name)

**********END**********

Developing Writing Skills

EX1

Objective: to have students examine logical sentence order

TEACHER:

Type several paragraphs. In each one, move one or two sentences so that they're out of sequence. Choose passages in which actions occur in a related sequence so that transitions (then, after that, etc.) give students clues as to the right order.

STUDENT ACTIVITY

Rearrange the following sentences so that the paragraph makes sense.

All of the lights in the house were out. Everyone was asleep, except for detective Phyllis Harlow. She sat in the dark at the top of the stairs, clutching her flashlight. It was just the grandfather clock in the library, striking midnight. Then the quiet of the house was broken by a loud BONG. Phyllis jumped at the sudden noise, and flashed a light down into the hallway. She turned off her flashlight and settled down again. When the door creaked open, she thought it was part of her dream. Then she heard footsteps and she shot up, wide awake. She began to doze off.

**********END**********

TEACHER:

You can extend this exercise later to explore logical paragraph order. Type a short essay, such as a newspaper story, encyclopedia article, or a long joke. Place the paragraphs in a nonsensical sequence, and ask students to rearrange them to make sense.

EX2

Objective: to give students practice in creating paragraphs using topic sentences

TEACHER:

Type a long passage, such as a short encyclopedia article or a student report. As you type, merge the paragraphs. If the piece has clear topic sentences, you may wish to remove them and ask students to insert topic sentences. An example of a long passage is shown below.

STUDENT ACTIVITY

Divide this essay into four paragraphs. Decide where each paragraph should begin and end. Press RETURN and indent four spaces to create paragraphs. Then insert a topic sentence to begin each paragraph.

Mark Twain was a famous American author of the 19th century. His greatest book, THE ADVENTURES OF HUCKLEBERRY FINN, has been called the first American classic. Twain's real name was Samuel Clemens, and he was born in 1835. He grew up in Hannibal, Missouri, on the banks of the Mississippi River. That river would play an important part in many of his later books. His pen name, Mark Twain, came from the river—it was a term used by riverboat pilots when they measured the river's depth. Mark Twain began writing after the Civil War. He started as a newspaper journalist in the "Wild West." Later,

EX6

Objective: to have students explore how transitional phrases shape stories and essays

TEACHER:

Type a series of transitional phrases or adverbs, followed by blanks, to use as sentence starters. Each series might be chosen to demonstrate a different kind of logical sequence (such as chronological order, physical movement, order of factual importance, and so on).

STUDENT ACTIVITY

Beginning in the first blank, type the rest of a sentence that starts telling a story. Continue the story, and try to make your plot lead you to a point where you can use the next sentence starter. Work your way through all the sentence starters to the end of the story.

First, . . .
Then, . . .
After that, . . .
Finally, . . .

**********END**********

TEACHER:

Below are further examples of sentence starters you could use:

Far away, . . .
Coming closer, . . .
Closer still, . . .
Face to face, . . .
To begin with, . . .
In addition, . . .
Nevertheless, . . .
In spite of this, . . .
When all is said and done, . . .

EX7

Objective: to give students practice in the use of meter and rhyme

TEACHER:

Type the instructions and starter sentences below. Later you can change the starters. Choose words that have many rhymes; create a line of verse ending with each word.

```
)RUN
PLEASE STANDBY - LOADING WORD LISTS
The discoverv of the planet Uranus bv William Herschel in 1781 and measurements
 its distance from the sun seemed to lend credence to the formula of planetary s
cings known as Bode's law (
 33 )

 1   2   3   4   5   6   7   8   9   10   11   12   13   14
 15   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30
1   32   33
AFTER to
the(*) discoverv(ADV) of(PREP) the(*) planet() Uranus(N) bv(PREP) William(N) Her
hel(N) in(PREP) 1781(N) and(CJ) measurements(N) of(PREP) its(ADJ) distance(N) fr
(PREP) the(*) sun() seemed(VP) to(INFIN) lend(VP) credence(N) to(PREP) the(*) fo
ula() of(PREP) planetarv() spacings(GERUND)) known() as(PREP) Bode's(ADJ) law(N)
AFTER PREP. PHRASES
the(*) discoverv(ADV) of(PREP) the(*) planet(N) Uranus(N) bv(PREP) William(N) He
chel(N) in(PREP) 1781(N) and(CJ) measurements(N) of(PREP) its(ADJ) distance(N) f
m(PREP) the(*) sun(N) seemed(VP) to(INFIN) lend(VP) credence(N) to(PREP) the(*)
rmula(N) of(PREP) planetarv(N) spacings(GERUND) known() as(PREP) Bode's(ADJ) law
)
AFTER ARTICLES
the(*) discoverv(ADV) of(PREP) the(*) planet(N) Uranus(N) bv(PREP) William(N) He
chel(N) in(PREP) 1781(N) and(CJ) measurements(N) of(PREP) its(ADJ) distance(N) f
m(PREP) the(*) sun(N) seemed(VP) to(INFIN) lend(VP) credence(N) to(PREP) the(*)
rmula(N) of(PREP) planetarv(N) spacings(GERUND) known() as(PREP) Bode's(ADJ) law
)
AFTER PATTERNS - FINAL PRODUCT
the(*) discoverv(ADV) of(PREP) the(*) planet(ADJ) Uranus(N) bv(PREP) William(N)
rschel(N) in(PREP) 1781(N) and(CJ) measurements(N) of(PREP) its(ADJ) distance(N)
rom(PREP) the(*) sun(N) seemed(VP) to(INFIN) lend(VP) credence(N) to(PREP) the(*
formula(N) of(PREP) planetarv(N) spacings(GERUND) known(N) as(PREP) Bode's(ADJ)
w(N)

The formula also predicted the existance of a planet between the orbits of Mars
d Jupiter so that manv astronomers undertook to find it ( 24 )

SUBORDINATE ADVERBIAL PHRASE...
so that manv astronomers undertook to find it
 1   2   3   4   5   6   7   8   9   10   11   12   13   14
 15   16   17   18   19   20   21   22   23   24
AFTER to
the(*) formula() also(ADV) predicted(V) the(*) existance(N) of(PREP) a(*) planet
 between(PREP) the(*) orbits(ADJ) of(PREP) Mars(N) and(CJ) Jupiter(N) so(ADV) th
(R-PRO) manv(ADJ) astronomers(N) undertook(V) to(INFIN) find(V) it(PRO)
AFTER PREP. PHRASES
the(*) formula() also(ADV) predicted(V) the(*) existance(N) of(PREP) a(*) planet
) between(PREP) the(*) orbits(ADJ) of(PREP) Mars(N) and(CJ) Jupiter(N) so(ADV) t
t(R-PRO) manv(ADJ) astronomers(N) undertook(V) to(INFIN) find(V) it(PRO)
AFTER ARTICLES
the(*) formula(N) also(ADV) predicted(V) the(*) existance(N) of(PREP) a(*) plane
N) between(PREP) the(*) orbits(ADJ) of(PREP) Mars(N) and(CJ) Jupiter(N) so(ADV)
at(R-PRO) manv(ADJ) astronomers(N) undertook(V) to(INFIN) find(V) it(PRO)
AFTER PATTERNS - FINAL PRODUCT
the(*) formula(N) also(ADV) predicted(V) the(*) existance(N) of(PREP) a(*) plane
N) between(PREP) the(*) orbits(ADJ) of(PREP) Mars(N) and(CJ) Jupiter(N) so(ADV)
at(R-PRO) manv(ADJ) astronomers(N) undertook(V) to(INFIN) find(V) it(PRO)

In 1801 the first and largest asteroid Ceres was discovered at about the right d
tance ( 15 )
```

BEST COPY AVAILABLE